

REMARKS

Claims 1 - 32 were pending and rejected. In response, claims 1, 3-4, 10, 14, 19-20, 22-23, 26, and 28-30 are amended and claims 2, 16, 21, and 27 are canceled. Claims 1, 3-15, 17-20, 22-26, and 28-32 are pending upon entry of this amendment. These changes are believed not to introduce new matter, and their entry is respectfully requested. In view of the Amendments herein and the Remarks that follow, Applicant respectfully requests that Examiner reconsider all outstanding objections and rejections, and withdraw them.

Double Patenting

Claims 2, 21, and 27 were provisionally rejected on the grounds of nonstatutory obviousness-type double patenting over claims 13 and 41 in copending application 10/642,355 ('355) in view of Sobel, U.S. Patent 6,205,558. This rejection is respectfully traversed.

Claim 1, as amended, incorporates claim 2, reciting a computer implemented method for rolling back a system state after a modification failure, the method comprising the steps of:

- a rollback manager creating a restore point on a computer;
- the rollback manager storing a reboot indicator in non-reversible storage;
- the rollback manager monitoring the reboot indicator to detect an unexpected reboot during deployment of a modification, the monitoring comprising:
 - the rollback manager detecting a reboot of the computer; and
 - the rollback manager determining based at least in part on the reboot indicator whether the reboot was expected or whether the reboot was unexpected; and
- responsive to determining that at least one unexpected reboot occurred during the deployment of the modification, the rollback manager rolling back the system state of the computer according to the restore point.

Claims 20 and 26 contain similar language to claim 1, and all arguments presented below regarding claim 1 equally apply to these claims.

Prior to the amendment herein, claim 2 was dependent on claim 1 and therefore included all of the limitations of claim 1. Applicant respectfully submits that Examiner did not properly consider the limitations of claim 1 when making the obviousness-type double patenting rejection of claim 2. Claim 2 cannot be obvious if its base claim is not obvious.

As amended herein, claim 2 is canceled and incorporated into claim 1. Amended claim 1 is not obvious over claim 13 of '355 in view of Sobel. Claim 13 of '355 mentions that a remediation action includes displaying a message and receiving a user input. Claim 13 of '355 depends on claim 1 of '355, which is concerned with rolling back a computer to a secure state and performing a remediation action if the state is not secure. Neither claim 13 of '355 nor Sobel mentions determining whether a reboot was expected or unexpected. Examiner states in the Office Action that "'355 claim 13 **also recites** method for roll-back via managing rollback with determination whether rollback is secure, and rolling back responsive to a state being managed or monitored" (emphasis added). However, claim 1 of the present application does not recite this language or anything similar to it.

Response to Rejection Under 35 USC § 112, Paragraph 2

Claims 14-16 were rejected under 35 USC § 112, ¶ 2 as allegedly being indefinite because one of ordinary skill in the art would not know whether the "reading the reboot indicator" in claim 14 is prior to or subsequent to a reboot of the computer. Applicant has amended claim 14 to clarify that the reading is performed after a reboot of the computer and before a booting of an operating system. Applicant submits that claim 14, as amended, is definite. Support in the specification for the amendment is found, for example on page 7, line 21 to page 8, line 6. Applicant respectfully requests that Examiner withdraw the rejection to claim 14 and dependent claims 15-16.

Response to Rejection Under 35 USC 103(a)

Claims 1-32 were rejected under 35 USC § 103(a) as allegedly being unpatentable over Sobel, U.S. Patent No. 6,205,558. This rejection is respectfully traversed.

Claim 1, as amended, incorporates claim 2, reciting a computer implemented method for rolling back a system state after a modification failure, the method comprising the steps of:

- a rollback manager creating a restore point on a computer;
- the rollback manager storing a reboot indicator in non-reversible storage;
- the rollback manager monitoring the reboot indicator to detect an unexpected reboot during deployment of a modification, the monitoring comprising:
 - the rollback manager detecting a reboot of the computer; and
 - the rollback manager **determining based at least in part on the reboot indicator whether the reboot was expected or whether the reboot was unexpected;** and
- responsive to determining that at least one unexpected reboot occurred during the deployment of the modification, the rollback manager rolling back the system state of the computer according to the restore point.

(emphasis added)

As can be seen, the claim recites creating a restore point on a computer and storing a reboot indicator in non-reversible storage. The reboot indicator is monitored to detect a reboot during a deployment of a modification. Based at least in part on the reboot indicator, it is determined whether the reboot was expected or unexpected. Responsive to determining that at least one unexpected reboot occurred, the system state of the computer is rolled back according to the restore point. The claimed invention can beneficially be used to detect an unexpected reboot during deployment of a modification that may result in the computer being left in an unknown or undesired system state. The system state of the computer can then be rolled back according to the restore point. Support in the specification is found, for example, on page 6, line 6 to page 9, line 8.

Claims 20 and 26 contain similar language to claim 1, and all arguments presented below regarding claim 1 equally apply to these claims.

Claim 1 is not obvious in view of Sobel. Sobel discloses a system for modifying a File Allocation Table (FAT) of a file system. In Sobel, the Master Boot Record (MBR) 110, a region of the disk that is automatically run after a reboot, is temporarily replaced with a recovery program 130 while the FAT is being modified. As a result, if the computer reboots during the modification of the FAT, the recovery program 130 will run. The recovery program can then take various actions, such as re-attempting the modification or restoring the original version of the FAT.

While Sobel discloses restoring the FAT to the original version upon a reboot, Sobel does not disclose determining whether the reboot was expected or unexpected and rolling back the system state as a result of an unexpected reboot. A reboot is a normal and expected part of certain kinds of modifications, and a rolling back of the system state may not be desired after an expected reboot. Sobel does not distinguish between expected and unexpected reboots. The system in Sobel always runs the recovery program 130 if a reboot occurs during a modification. Additionally, the system in Sobel does not monitor a reboot indicator but rather rewrites the MBR of a disk so that the portion is automatically run after a reboot.

Accordingly, Sobel does not disclose “the rollback manager determining based at least in part on the reboot indicator whether the reboot was expected or whether the reboot was unexpected.” Rather, the recovery program in Sobel is run after any reboot during a modification (Sobel, col. 5, lines 30-32). Sobel is not concerned with the possibility of a modification involving an expected reboot.

Based on the above remarks, Applicant respectfully submits that for at least these reasons claims 1, 20, and 26 are patentably distinguishable over the cited reference. Therefore,

Applicant respectfully requests that Examiner reconsider the rejection and withdraw it. As to dependent claims, because claims 3-19, 22-25, and 28-31 variously depend on claims 1, 20, and 26, all arguments advanced above with respect to claims 1, 20, and 26 are hereby incorporated so as to apply to these dependent claims.

Claim 32 recites a computer implemented method for auditing a computer system state, the method comprising the steps of:

- a rollback manager auditing the computer and storing in non-reversible storage information concerning at least one item from a group of items consisting of:
 - at least one **currently executing system process**;
 - at least one **currently executing user process**; and
 - at least one **currently open listening port**.

(emphasis added)

As can be seen, the claim recites auditing a computer and storing information concerning a currently executing system process, a currently executing user process, or a currently open listening port. Such an audit can be beneficially used to determine if a modification was successful. For example, audit information stored prior to a modification can be compared to the system state after a modification to determine if the modification was successful. Support in the specification is found, for example, on page 9, line 20 to page 10, line 13.

Dependent claims 3, 22, and 28 contain similar language to claim 32, and all arguments presented below regarding claim 32 equally apply to these claims.

Sobel does not mention any of a system process, a user process, or an open listening port. As a result, Sobel does not disclose storing information about any of these. In rejecting claim 32, Examiner refers to the rejection of claim 3, where Sobel, col. 4, lines 26-37, is cited as disclosing storing information about at least one currently executing user process. However, this portion of Sobel merely mentions that a user can reboot the computer upon a failed modification attempt. Examiner further mentions that “event based user input reads on user

process.” However, the Examiner misunderstands the meaning of the term “process” in the claimed invention. The words of a claim must be given their “plain meaning” unless such meaning is inconsistent with the specification, where “plain meaning” refers to the ordinary and customary meaning given to the term by those of ordinary skill in the art. See MPEP 2111.01. A “process” is a term known to a person of ordinary skill in the art and refers to an instance of a computer program that is being sequentially executed on a computer. A “user process” is a process associated with a user of the computer. Sobel does not disclose storing information concerning a currently executing user process as in the claimed invention.

Based on the above remarks, Applicant respectfully submits that for at least these reasons claims 3, 22, 28, and 32 are patentably distinguishable over the cited reference. Therefore, Applicant respectfully requests that Examiner reconsider the rejection and withdraw it.

Conclusion

Accordingly, Applicant respectfully requests allowance of this application. The Examiner is invited to contact the undersigned by telephone to advance the prosecution of this case.

Respectfully submitted,
WILLIAM E. SOBEL

Dated: December 27, 2007

By: /Nikhil Iyengar/
Nikhil Iyengar, Reg. No. 60,910
Fenwick & West LLP
Silicon Valley Center
801 California Street
Mountain View, CA 94041
Tel.: (650) 335-7627
Fax: (650) 938-5200